REMARKS/ARGUMENTS

Claims 1-27 are pending in the present application and have been finally rejected.

Claim Rejections

The Examiner rejected claims 1-26 under 35 U.S.C. §103(a) as being unpatentable over

Tanaka et al. (U.S. Patent No. 5,542,064) in view of Microsoft Computer Dictionary and further

in view of Applicants' Admitted Prior Art (AAPA). In rejecting the independent claims, the

Examiner stated:

> Regarding claims 1, 3, 5, 11, 13, 15, 20 and 22, Tanaka et al. teaches a data storage system with a plurality of storage devices (disk drives 16-1 to 16-n, Figure 1) in which CPU 1 is the main module of which controller 2 is a child, controller 2 is a module of which the plurality of disk processors 17-1 to 17-n are children, and the plurality of disk processors are independent modules each having a disk drive 16-1 to 16-n as a child (Figure 1). Tanaka discloses controller 2 receiving input and output command from CPU 1 and passing such commands from the controller 2 to the corresponding drive processor 17-1 to 17-n and then to the corresponding disk drive ("deciding which child to pass the input command to … passing the input command to the decided child", Column 3, line 50 to Column 4, line 35). In this system, the source is transparent to the drive processor module in that this module does not communicate with the CPU, but instead communicated directly with a module above it, controller 2. Tanaka does not teach each module comprising programming code for implementing a RAID storage system. Microsoft Computer Dictionary discloses a RAID storage system (Page 372). I would have been obvious to one of ordinary skill in the art at the time the invention was made to transform the system of Tanaka et al to include a RAID storage system since a RAID system provides high performance, storage efficiency, high speeds, reliability and low cost. **Although Tanaka et al. discloses "no consideration of the input/output command waiting state of each disk drive at the time of reading/writing of data" as a disadvantage of a RAID system (Column 2, lines 14-20), the many advantages of using a RAID system might be a tradeoff valuable enough to still use this system despite its small disadvantage.** Tanaka in view of Microsoft Computer Dictionary does not teach encapsulated modules where inputs and outputs are not fixed and modules can be mixed and matched to form different RAID configurations. Applicants Admitted Prior Art discloses an encapsulated RAID system (Figure 2B) where RAID 5 modules are encapsulated within a RAID 0 module. It would have been obvious to one of ordinary skill in the art at the time the invention was made to adapt the encapsulation of the admitted prior art to the system of Tanaka in view of Microsoft Computer Dictionary since such encapsulation isolates modules within modules thus creating more manageable sub-systems within the main system.

Applicants respectfully disagree.

The present invention is directed to a method and system for implementing non-hierarchical and hierarchical RAID configurations. According to the preferred embodiment of the present invention, non-hierarchical and hierarchical RAID configurations are viewed as a combination of RAID modules. Each particular RAID module is encapsulated, i.e., inputs and outputs are not fixed to any particular device. Thus, any hierarchical RAID configuration and any number of hierarchical tiers can be built by combining the encapsulated RAID modules. Each module represents a particular RAID level, in other words, each module contains programming code representing the features of a particular RAID level. Each encapsulated module is adapted to receive an input command, which is related to the data stored in the physical storage devices. Unlike the conventional system, however, the module is not coded to *assume* that the source of the input is the host. Thus, in the preferred embodiment of the present invention, the source of the input command is transparent to the module.

In addition, each encapsulated module includes at least one output path, or child. The module is *not* coded to *assume* that its children are disk drives. Thus, the destination of each child is transparent to the module. Because each module is encapsulated, programming modules can be mixed and matched to form hierarchical RAID configurations with little or no programming effort. As a result, the code is simpler and easier to test and debug.

The present invention provides, as recited in claim 1:

> 1. A method for managing data in a data storage system, the data storage system including a plurality of physical storage devices, the method comprising the steps of:
> a) providing a plurality of encapsulated modules, wherein each module comprises programming code for implementing features of a storage system utilizing a Redundant Array of Inexpensive Disks (RAID) configuration and each module comprises at least one child;
> b) receiving an input command related to the data by one of the

3

plurality of modules from a source, wherein the source is transparent to the one module;

      c)   deciding which child of the at least one children to pass the input command; and

      d)   passing the input command to the decided child for processing the data according to the input command.

Claims 11 and 20 are computer readable medium and system claims having similar scopes to claim 1.

In contrast to the present invention, Tanaka is directed to a method for controlling how data and back up data is written into a predetermined number of storage units, i.e., disks, in a secondary storage device. According to Tanaka, the secondary storage device 101 (Figure 2) includes a controller 2 that manages input and output commands and data from a CPU 1. The controller 2 selects disks 16-1 to 16-n having free capacity and a lower number of input/output commands to be processed, and writes multiple copies of data to those selected disks. The controller 2 maintains a plurality of tables 12 for performing this function. The tables include an address management table 50 for managing data stored in the respective disks, a free area management table 55 for managing free areas in the respective disks and a disk drive management table 60 for managing the number of input-output commands to be processed in each of the disks.

Microsoft Computer Dictionary at page 372 defines RAID as a "data storage method in which data, along with information used for error correction, . . . is distributed among two or more hard disks in order to improve performance and reliability. The hard disk array is governed by array management software and a disk controller, which handles the error correction. . . . Several defined levels of RAID offer differing trade-offs among access speed, reliability, and cost."

Finally, the AAPA, and in particular Figure 2B, describes a hierarchical RAID system

where the RAID 0 module is hard-coded to receive inputs from the host and to transmit outputs to one of the RAID 5 modules, and each RAID 5 module is hard-coded to receive inputs from the RAID 0 module and to transmit outputs to one of its disk drives. In other words, each module is aware of its source and also aware of the recipient of its commands. The configuration in Figure 2B and the accompanying text describe the difficulties associated with building hierarchical RAID systems, specifically the additional customized coding that must be added to each RAID module. See Specification at page 4, lines 12-21.

Tanaka in view of Microsoft in view of the AAPA fails to teach or suggest "providing a plurality of encapsulated modules, wherein each module comprising programming code for implementing features of a storage system utilizing a Redundant Array of Inexpensive Disks (RAID) configuration," as recited in claims 1, 11 and 20. In the present invention, encapsulated RAID modules are used as building blocks to create hierarchical RAID configurations. By taking an objected oriented approach, hierarchical and nonhierarchical RAID configurations are easily implemented without requiring new programming code for each configuration. Because particular RAID modules are encapsulated, i.e., inputs and outputs are not fixed to any particular device, any hierarchical RAID configuration and any number of hierarchical tiers can be built by combining the encapsulated RAID modules. Minimal or no additional programming is necessary. As an added feature, background functions, such as rebuilding a degraded disk drive, can also be encapsulated in a programming module, which can then be used to simplify coding, testing and debugging.

Applicants respectfully submit, and the Examiner concedes, that none of the modules disclosed in Tanaka, e.g., the CPU 1, the controller 2, or the disk processors, implement "features of a storage system utilizing a . . . RAID configuration," as recited in claims 1, 11 and 20. Indeed, Tanaka expressly criticizes the RAID configuration because "there is no consideration of

the input/output command waiting state of each disk drive at the time of reading/writing of data in a plurality of disk drives." (Column 2, lines 14-17). Nevertheless, despite Tanaka's explicit rejection of the RAID configuration, the Examiner states that it would be obvious to transform Tanaka's system "to include a RAID storage system since RAID system provides" performance advantages (as defined by Microsoft Computer Dictionary) and "the many advantages of using a RAID system *might be a tradeoff* valuable enough to still use this system despite its small disadvantage." Applicants respectfully disagree for several reasons.

First, neither Tanaka nor Microsoft provide explicit or implicit motivation to a person skilled in the art to combine the references in such a way as suggested by the Examiner. As stated above, Tanaka explicitly steers away from RAID because "there is no consideration of the input/output command waiting state of each disk drive at the time of reading/writing of data in a plurality of disk drives." (Column 2, lines 14-17). The Microsoft definition states "[s]everal defined levels of RAID offer *differing trade-offs* among access speed, reliability, and cost." The fact that RAID is *defined* in Microsoft Computer Dictionary does not teach or suggest that Tanaka can or should include RAID in the manner suggested by the Examiner, particularly in light of the above cited criticism. Accordingly, Applicants respectfully submit that one skilled in the art would not be motivated to combine the references as suggested by the Examiner.

Secondly, even if combined, the combination of Tanaka and Microsoft, i.e., transforming Tanaka's system "to include a RAID storage system" as the Examiner suggests, still fails to teach or suggest the present invention, as recited in claims 1, 11 and 20. The combination still fails to teach or suggest managing the data by "providing a plurality of encapsulated modules, wherein *each* module comprising programming code for implementing features of a storage system utilizing a Redundant Array of Inexpensive Disks (RAID) configuration," as recited in claims 1, 11 and 20.

None of Tanaka's "modules," e.g., the CPU 1, the controller 2, or the disk processors are designed to implement RAID and therefore do not include "programming code for implementing features of a storage system utilizing a Redundant Array of Inexpensive Disks (RAID) configuration." The CPU 1, controller 2 and disk processors each have specific functions --- functions that *do not* implement RAID. Accordingly, Tanaka in view of Microsoft fails to mention or suggest a plurality of encapsulated modules that include "programming code for implementing features of a storage system utilizing a Redundant Array of Inexpensive Disks (RAID) configuration," as recited in claims 1, 11 and 20.

Moreover, Applicants respectfully submit that the combination of Tanaka, Microsoft, and the AAPA (in particular Figure 2B) fails to teach or suggest "a plurality of *encapsulated* modules" and receiving from a source an input command by one of the modules, "wherein the source is transparent to the one module." In the present invention, each RAID module is encapsulated, i.e., inputs and outputs are not fixed to any particular device. Thus, in the preferred embodiment of the present invention, the source of the input command is transparent to the module. By encapsulating the modules in this way, any hierarchical RAID configuration and any number of hierarchical tiers can be built without customized coding for each module.

None of the cited references teach this feature. In Tanaka, each "module" interacts and works in cooperation with the other "modules." For example, the CPU 1 transmits input and output commands and data to the controller 2. The CPU 1 is aware it is interacting with the controller 2 and the controller is aware that it is receiving commands and data from the CPU 1. The controller 2, in turn, selects a particular disk drive 16-1 based on various factors and transmits the commands and data to a drive processor 17-1 associated with the selected disk drive 16-1. The disk processor 17-1 is aware that it is receiving commands and data from the controller 2, which in this case is its source of the commands and data. Accordingly, none of

Tanaka's modules is "encapsulated" and none receive commands from a source that is "transparent" to the module.

In the Office Action, the Examiner notes that the drive processor 17-1 does not communicate with the CPU 1 and therefore, "the source is transparent to the drive processor module." Applicants respectfully submit, however, that the CPU 1 is not the "source" because the drive processor 17-1 does not receive the commands and data from the CPU 1, but rather from the controller 2. Thus, Applicants respectfully submit that the controller 2 is the source for the drive processor 17-1 and that the controller 2 is not transparent to the drive processor 17-1.

With respect to the AAPA, Applicants respectfully submit that each RAID module illustrated in Figure 2B is not encapsulated because each module is hard-coded to receive commands and data from a specific source and to pass commands and data to a particular entity. For example, the RAID 0 module is hard coded to receive inputs from the host and to transmit outputs to a RAID 5 module. (Specification, page 4, lines 12-21). Thus, the source is not "transparent" to the module and the modules in FIG. 2B cannot be reconfigured without changing the hard-code in each module. The Specification explicitly states that this is the disadvantage of the AAPA because each module requires additional hard-coding to implement the hierarchical configuration.

Applicants respectfully submit that Tanaka in view of Microsoft and further in view of the AAPA fails to teach or suggest the cooperation of elements recited in claims 1, 11 and 20. Accordingly, claims 1, 11 and 20 are allowable. Claims 2-10, 12-19 and 21-26 depend on claims 1, 11 and 20, respectively, and the above arguments apply with equal force. Therefore, Applicants respectfully submit that claims 2-10, 12-19 and 21-26 are also allowable.

With regard to claim 27, the Examiner does not indicate whether this claim is allowable. Applicants respectfully submit, however, that claim 27 depends on claim 20 and that the above
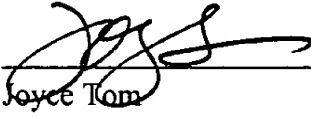
arguments apply with equal force. Accordingly, claim 27 is allowable over the cited references.

In view of the foregoing, it is submitted that the claims 1-27 are allowable over the cited references and are in condition for allowance. Applicants respectfully request reconsideration of the rejections and objections to the claims, as now presented.

Applicants believe that this application is in condition for allowance. Should any unresolved issues remain, Examiner is invited to call Applicants' attorney at the telephone number indicated below.

Respectfully submitted,

SAWYER LAW GROUP LLP

September 9, 2004
Date

Joyce Tom
Attorneys for Applicant(s)
Reg. No. 48,681
(650) 493-4540